

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑪ 公開特許公報(A) 平1-271861

⑫ Int.Cl.
G 06 F 13/28

識別記号 庁内整理番号
310 J-8840-5B

⑬ 公開 平成1年(1989)10月30日

審査請求 未請求 請求項の数 1 (全6頁)

⑭ 発明の名称 直接メモリ・アクセス制御装置

⑮ 特 願 昭63-100656

⑯ 出 願 昭63(1988)4月23日

⑰ 発 明 者 白 江 有 利 神奈川県大和市深見西4丁目2番49号 株式会社ビーエフ
ユー大和工場内

⑱ 出 願 人 株式会社ビーエフユー 石川県河北郡字ノ気町字野気ヌ98番地の2

⑲ 代 理 人 弁理士 京 谷 四 郎

明 細 書

1. 発明の名称

直接メモリ・アクセス制御装置

2. 特許請求の範囲

メモリ・アドレスを保持するメモリ・アドレスレジスタ部と、
転送バイト数をカウントするためのカウントレジスタ部と、
転送データを一時的に蓄えるデータ・バッファメモリ部と、
データ・バッファ・メモリの有効バイト数を制限するための有効バイト数制限レジスタ部と、
データ・バッファ・メモリ部をアクセスするためのメモリ・ポインタ部と
を具備し、
データ・バッファ・メモリ部の有効バイト数をバーストDMA転送の一回あたりのデータとし、その有効バイト数を可変とすることで、バースト

転送時間を変えられるようにしたことを特徴とする直接メモリ・アクセス制御装置。

3. 発明の詳細な説明

(概要)

内部にデータ・バッファ・メモリを持つDMA制御装置に関し、
有効バイト数を可変とすることによりバースト転送時間を短縮し得るようにすることを目的とし、
メモリ・アドレスを保持するメモリ・アドレスレジスタと、転送バイト数をカウントするためのカウント・レジスタと、転送データを一時的に蓄えるデータ・バッファ・メモリと、データ・バッファ・メモリの有効バイト数を制限するための有効バイト数制限レジスタと、データ・バッファ・メモリをアクセスするためのメモリ・ポインタとを具備し、データ・バッファ・メモリの有効バイト数をバーストDMA転送の一回あたりのデータとし、その有効バイト数を可変とすることで、バースト転送時間を変えられるようにしたもので

ある。

〔産業上の利用分野〕

本発明は、内部にデータ・バッファ・メモリを持つDMA制御装置（直接メモリ・アクセス制御装置）に関するものである。制御用コンピュータを始めとしてI/O装置との高速なデータ転送が必要な分野においてDMA転送は必須のものである。

〔従来の技術〕

第8図は従来のDMA制御方式のブロック図である。同図において、1はDMA制御装置、2はメモリ、3はI/O装置、4は中央処理装置、11はメモリ・アドレス・レジスタ、12はバイト・カウント・レジスタをそれぞれ示している。

第6図に示すように、従来のDMA制御方式は、メモリ・アドレスを保持するためのメモリ・アドレス・レジスタ11および転送バイト数をカウントするためのカウント・レジスタ12を持つDMA制御装置1と、メモリ2と、I/O装置3と、中央処理装置4とで構成されている。

この種の従来例においては、中央処理装置4がDMA制御装置1に対してメモリ・アドレスと転送バイト数とを設定して起動させると、カウント・レジスタ12の値が零になるまで転送を続ける。この際、バスを継続的に使用する方式（バースト転送方式）もしくは1ワード分のDMA転送を行う毎にバスを開放する方式（サイクル・スチール転送方式）の何れかの方式で行うのが一般的である。

〔発明が解決しようとする課題〕

前者の方式は、バス支配権制御に要する時間が小さくて済むため、高速且つ効率の良い転送が可能であるが、その反面、バスを長時間にわたって継続的に使用するため、その間の割込み要求へのレスポンスの遅れ等、システム全体のリアルタイム性への影響がある。

後者の方式は、リアルタイム性への影響はないが、バス支配権制御に要する時間が大きくなり過ぎると言う欠点がある。

本発明は、この点に鑑みて創作されたものである。

- 3 -

って、有効バイト数を可変とすることによりバースト転送時間を変化し得るようになったDMA制御方式を提供することを目的としている。

〔課題を解決するための手段とその作用〕

第1図は本発明のDMA制御装置を示すブロック図である。同図において、1はDMA制御装置、11はメモリ・アドレス・レジスタ(MAR)、12はバイト・カウント・レジスタ(BCR)、13は有効バイト数制限レジスタ、14はメモリ・ポインタ・レジスタ(MPR)、15はデータ・バッファ・メモリ、17と18はデータバスをそれぞれ示している。DMA制御装置1は、メモリ・アドレス・レジスタ11、バイト・カウント・レジスタ12、有効バイト数制限レジスタ13、メモリ・ポインタ・レジスタ14及びデータ・バッファ・メモリ15などを有している。メモリ・アドレス・レジスタ11は、データバス17に接続されているメモリをアクセスするためのアドレスを保持する。バイト・カウント・レジスタ12は、転送バイト数をカウントするものである。データ・バッ

- 4 -

ファ・メモリ15は、転送データを一時的に蓄えるものである。有効バイト数制限レジスタ13は、データ・バッファ・メモリ15の有効バイト数を制限するためのものである。メモリ・ポインタ・レジスタ14は、データ・バッファ・メモリ15をアクセスするための番地を保持するものである。

例として、I/O装置からメモリへDMA転送する場合の動作を説明する。I/O装置からデータバス18を経由してDMA制御装置1に入力されたデータは、メモリ・ポインタ・レジスタ14の指す番地のデータ・バッファ・メモリ15に次々とストアされる。データ・バッファ・メモリ15内のデータ量が有効バイト数制限レジスタ13の指示するバイト数に達すると、データ・バッファ・メモリ15内のデータは、データバス17を経由してアドレス・バス16でアドレスシンクされるメモリ番地にバースト転送で次々と転送される。この場合、有効バイト数制限レジスタ13の値を大きくすればデータ・バッファ・メモリ内に一度に蓄えるバイト数が大きくなるのでバースト

- 5 -

- 392 -

- 6 -

転送時間が長くなり、逆に小さくすればバースト転送時間は短くなる。その頃はシステムに応じて選べば良い。有効バイト数制限レジスタ13の値は、中央処理装置4によって書替可能である。

〔実施例〕

第2図は有効バイト数オーバーの検出を行うハードウェア機構の構成例のブロック図である。同図において、19はデコーダ、20と21はAND回路、22はOR回路をそれぞれ示している。

有効バイト数制限レジスタ13の値をデコーダ19によってデコードした格号を使ってメモリ・ポインタ・レジスタ14の有効バイトを選択する。例えば、AND回路20は有効バイト数が2・バイトをオーバーすると“真”になり、AND回路21は有効バイト数の2ⁿ・バイトをオーバーすると“真”となり、それぞれOR回路22の出力を“真”とする。例えば、I/O装置からメモリへDMA転送を行う場合、I/O装置からのデータをデータ・バッファ・メモリ15にストアする毎にメモリ・ポインタ・レジスタ14の値をインク

リメントし、OR回路22の出力が“真”になれば、データ・バッファ・メモリ15にストアされたデータをメモリにバースト転送する。この動作をカウント・レジスタ12が零になるまで繰り返す。メモリからI/O装置への転送も同様に制御出来る。

第3図は本発明で使用者されるシーケンサの構成例を示す図である。同図において、23はカウンタ、24はマイクロコード・メモリ、25はマルチプレクサをそれぞれ示している。

第3図のシーケンサは、第1図には示されていないが、DMA制御装置1の中に存在し、DMA制御装置1の各部を制御する。カウンタ23は通常はカウント値を+1するが、分岐アドレス入力制御信号がオンのときには、読み出されたマイクロコードの分岐アドレスがカウンタ23にセットされる。マイクロコード・メモリ24からは、カウンタ23で指定された番地のマイクロコードが読み出される。マルチプレクサ25は、読み出されたマイクロコードの分岐条件選択部の内容で指

- 7 -

定された分岐条件が成立しているか否かを調べ、指定された分岐条件が成立している場合は、分岐アドレス入力制御信号をオンとする。

第4図はI/Oからメモリへデータを転送する場合の処理フローを示す図である。なお、これらの処理はシーケンサの制御の下で行われる。

- ① 中央処理装置がDMA制御装置(DMAC)およびI/O装置を起動する。
- ② I/O装置がDMA制御装置に対するデータ・リクエストDREQをオンするまで待つ。
- ③ DMA制御装置がI/O装置のデータ・レジスタ・ポートを選択してI/O装置からデータ・バッファ・メモリへのデータ転送を行う。
- ④ DMA制御装置はデータ・バッファ・メモリ内のデータのバイト数が有効バイト数をオーバーしたか否かを調べ、Yesの場合は⑤に進み、Noの場合は②に戻る。
- ⑤ DMA制御装置は、システム・バスを獲得するための要求を出す。
- ⑥ バスを獲得したならばデータ・バッファ・メ

- 8 -

モリ・メモリへのデータ転送を行う。(データ・バッファ・メモリが空になるまで続ける。)

- ⑦ 転送カウンタ(バイト・カウンタ・レジスタ同義)が0か否かを調べる。Noの場合は①の処理に戻る。

第5図はメモリからI/Oへデータ転送する場合の処理フローを示す図である。なお、これらの処理はシーケンサの制御の下で行われる。

- ① 中央処理装置がDMA制御装置およびI/O装置を起動する。
- ② DMA制御装置はシステム・バスを獲得するための要求を出す。
- ③ バスを獲得したならば、メモリ・データ・バッファ・メモリへのデータ転送を行う。
- ④ DMA制御装置は、データ・バッファ・メモリ内のデータのバイト数が有効バイト数をオーバーしたか否かを調べ、Yesの場合は⑤に進み、Noの場合は②に戻る。
- ⑤ I/O装置がDMA制御装置に対するデータ・リクエストDREQをオンにするまで待つ。

- ④ DMA制御装置がI/O装置のデータ・レジスタ・ポートを選択してデータ・バッファ・メモリからI/O装置へのデータ転送を行う。
- ⑤ ④の処理をデータ・バッファ・メモリが空になるまで繰り返す。
- ⑥ 転送カウンタが"0"か否かを調べ、Nのときは④の処理に戻る。
- (発明の効果)

以上の説明から明らかなように、本発明のDMA制御装置は、DMA転送時に転送データを一時的に内部のデータ・バッファ・メモリに取り込む機構を持っており、このデータ・バッファ・メモリのデータ・サイズを一かたまりとして、バーストのDMA転送を行う。本発明が従来技術と異なるのは、データ・バッファ・メモリの有効バイト数を可変とすることで、バースト転送時間を可変としたことにある。

4. 図面の簡単な説明

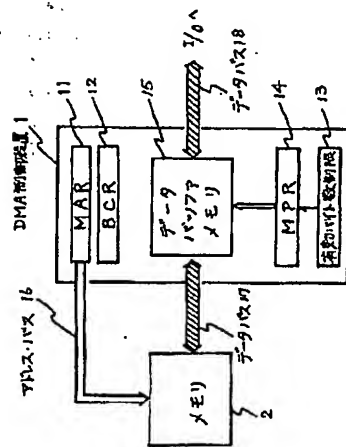
第1図は本発明のDMA制御装置を示す図、第

2図は有効バイト数オーバーの検出を行うハードウェア機構の構成例のブロック図、第3図は本発明で使用するシーケンサの構成例を示すブロック図、第4図はI/Oからメモリへデータを転送する場合の処理フローを示す図、第5図はメモリからI/Oへデータを転送する場合の処理フローを示す図、第6図は従来のDMA制御方式を示す図である。

1…DMA制御装置、2…メモリ、3…I/O装置、4…中央処理装置、11…メモリ・アドレス・レジスタ、12…バイト・カウンタ・レジスタ、13…有効バイト数制限レジスタ、14…メモリ・ポインタ・レジスタ、15…データ・バッファ・メモリ、17と18…データバス、19…デコーダ、20と21はAND回路、22…OR回路。

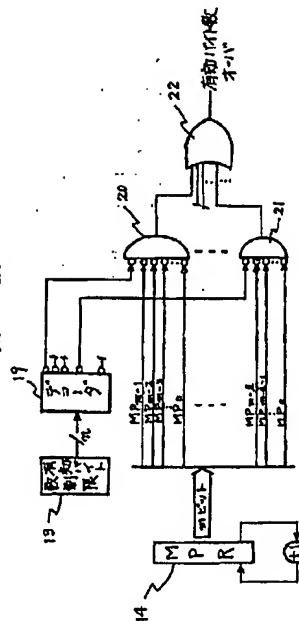
特許出願人 株式会社ビーエフユー
代理人弁理士 京谷 四郎

- 11 -

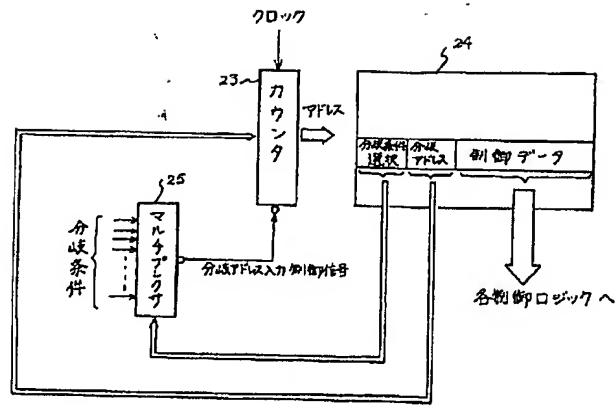


本発明のDMA制御装置
第1図

- 12 -

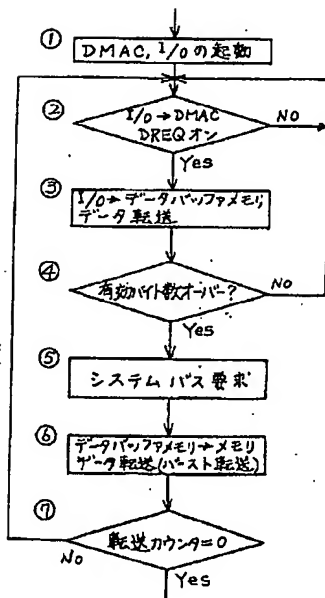


有効バイト数オーバーの検出
第2図



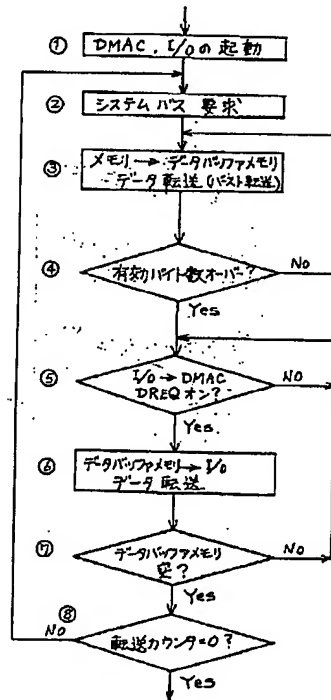
シーケンサの構成例

第3図



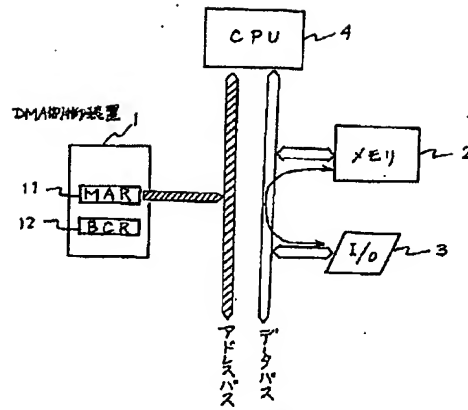
I/Oからメモリへのデータ転送

第4図



メモリからI/Oへのデータ転送

第5図



従来のDMA制御方式

第6図

J JAPAN PATENT OFFICE (JP)

J Unexamined Patent
Publication

J Publication of an Unexam-
ined Patent Application(A)

Hei 1-271861

(51) Int. Cl. ⁴	Identifica- tion Codes	JPO File Num- ber	(24) Publication Date
G06F 13/28	310	J-8840-5B	October 30, 1989

J

Number of Claims

1 (Total of 6 pages)

(54) DIRECT MEMORY ACCESS CONTROLLER

(21) Japanese Patent Application 63-100656

(22) Application Date April 23, 1988

(72) Inventor Yuri SHIRAE PFU Co., Ltd., Yamato Plant 2-49, Fu-
kami-nishi 4-chome, Yamato-shi, Kanagawa

(73) Patentee: PFU Co., Ltd. Nu 98-2 Unoke, Kahoku-shi, Ishikawa-ken

(74) Agent Patent Attorney Shiro KYOTANI

SPECIFICATION

1. Title of the Invention

DIRECT MEMORY ACCESS CONTROLLER

2. What is Claimed is:

A direct memory access controller comprising:

a memory address register (11) that holds memory addresses;
a count register (12) to count the number of bytes transferred;
a data buffer memory (15) to store transferred data temporarily;
a valid byte number limiting register (13) to limit the number of
valid
bytes of the data buffer memory; and
a memory pointer (14) to access the data buffer memory (15),
wherein said access controller is configured so that the valid byte
number of the data buffer memory (15) is a single burst DMA transfer,
the number of valid bytes is variable, and the burst transfer time is
variable.

3. Detailed Description of the Invention

(Summary)

The invention relates to DMA controllers having internal data buffer memory, with the objective of obtaining and variable burst transfer times by making the number of valid bytes and variable, and comprises a memory address register that holds memory addresses, a count register to count the number of bytes transferred, a data buffer memory to store transferred data temporarily, a valid byte number limiting register to limit the number of valid bytes of the data buffer memory, and a memory pointer to access the data buffer memory, and is configured so that the valid byte number of the data buffer memory is a single burst DMA transfer, the number of valid bytes is variable, and the burst transfer time is variable.

Industrial Field of Use

This invention relates to DMA controllers having internal data buffer memory (direct memory access controllers). Starting with control computers, DMA transfer is necessary in fields where high-speed data transfer with I/O devices is needed.

Prior Art

Figure 6 is a block diagram illustrating a prior art DMA control method. In this figure, 1 is the DMA controller, 2 is the memory, 3 is the I/O device, 4 is the CPU, 11 is the memory address register, 12 is the byte count register.

As shown in Figure 6, prior art DMA control methods comprise a DMA controller 1 having a memory address register 11 for holding memory addresses and a count register 12 for counting the number of bytes transferred; a memory 2; an I/O device 3; and a CPU.

In this type of prior art, when the CPU 4 is set for the memory address and the number of transfer bytes relative to the DMA controller 1 and is started, the transfer will continue until the value of the count register 12 reaches zero. Thus, the typical methods are either continuous use of the bus (burst transfer method), or the method in which the bus is released when a single word's worth of a DMA transfer is performed (cycle steal transfer method).

Problems the Invention Seeks to Resolve

Although the former method enables high-speed and highly efficient transfers since the time required to control the bus is small, the method conversely has a negative effect on the real-time characteristics of the overall system because there are delays caused by interrupt requests to the register and other problems due to long periods of continuous use of the bus.

With respect to the latter method, although there are no adverse effects on real-time characteristics, the method is defective because a large amount of time is required to control the bus.

In light of these points, it is therefore an objective of the present invention to provide a DMA controller that obtains variable burst transfer times by making the number of valid bytes variable.

Means of Solving the Problems and Operation of the Invention

Figure 1 shows a block diagram of the DMA controller of this invention. In this same figure, 1 is the DMA controller, 11 is the memory address register (MAR), 12 is the byte count register (BCR), 13 is

the valid byte number limiting register, 14 is the memory pointer register (MPR), 15 is the data buffer memory, and 17 and 18 are data buses. The DMA controller 1 comprises a memory address register 11, a byte count register 12, a valid byte number limiting register 13, a memory pointer register 14, and a data buffer memory 15, among others. The memory address register 11 holds addresses in order to access memory connected to the data bus 17. The byte count register 12 counts the number of bytes transferred. The data buffer memory 15 temporarily stores the transferred data. The valid byte number limiting register 13 is it to limit the number of valid bytes of the data buffer memory 15. The memory pointer register 14 holds addresses to access the data buffer memory 15.

The following example is an explanation of operations when there is a DMA transfer from an I/O device to memory. Data that is input from an I/O device to the DMA controller via data bus 18 is stored sequentially in the addresses of the data buffer memory 15 that are indicated by the memory pointer register 14. When the quantity of data in the data buffer memory 15 reaches the number of bytes indicated by the bite number limiting register 13, the data in the data buffer memory 15 is transferred sequentially via the data bus 17 by burst transfer to the memory addresses that have been addressed by the address bus 16. In this case, if the value of the valid byte number limiting register 13 is large, the number of bytes at one time in the buffer memory becomes large and the burst transfer time becomes longer. Conversely, if this value is small, the burst transfer time becomes shorter. This value can be selected according to the system. The value of the valid byte number limiting register 13 can be written according to the CPU 4.

Preferred Embodiments

Figure 2 is a block diagram of a configuration of a hardware mechanism for detecting the overage of valid bytes. In the figure, 19 is a decoder, 20 and 21 are AND circuits, and 22 is an OR circuit.

Signals that are decoded by the decoder 19 are used and the value of the valid byte number limiting register 13 selects the valid bytes

of the memory pointer register 14. For example, the AND circuit 20 becomes "true" when the number of valid bytes exceeds 2^n bytes, and the AND circuit 21 before becomes "true" when the number of valid bytes exceeds 2^{n-t+1} [partly illegible], and the output of the OR circuit 22 is "true" in either case. For example, when a DMA transfer is performed from the I/O device to memory, and the value of the memory pointer register 14 for each storage of data from the I/O device to the data buffer memory 15 is incremented, if the output of the OR circuit 22 is "true," the data stored in the data buffer memory 15 is burst-transferred to memory. This operation is repeated until the count register 12 goes to zero. Transfers from memory to I/O devices can similarly be controlled.

Figure 3 is a block diagram of a configuration of a sequencer used in this invention. In the figure, 23 is a counter, 24 is microcode memory, and 25 is a multiplexer.

Although not indicated in Figure 1, the sequencer in Figure 3 exists within the DMA controller, and it controls each part of the DMA controller I. The counter 23 typically increases the count by value of +1, but when the branch address input to control signal is on, the counter 23 sets the microcode branch address that is read out. The microcode address that is specified at the counter 23 is read out from microcode memory 24. The multiplexer 25 checks the read-out microcode branch conditions selection part content to determine whether or not branch conditions have been established, and turns the branch address input control signal on if the specified branch conditions have been established.

Figure 4 is a flow chart showing the processing when data is transferred from I/O to memory. This processing is performed by sequencer control as follows:

- ① The CPU starts the DMA controller (DMAC) and the I/O device.
- ② The I/O device waits until the data request DREQ to the DMA controller is turned on.

- ③ The DMA controller selects the I/O device data register port and performs a data transfer from the I/O device to the data buffer memory.
- ④ The DMA controller checks to see if the number of data bytes in the data buffer memory exceeds the number of valid bytes or not. If it is "yes", proceed to ⑥; if "no", return to ②.
- ⑤ The DMA controller issues a request to acquire the system bus.
- ⑥ When the bus has been acquired, data transfer from the data buffer memory → memory is performed. (Continues until the data buffer memory has been emptied.)
- ⑦ Check to see if the transfer counter (same as the byte counter register) is 0 or not. Return to process ② if it is "no".

Figure 5 is a flow chart showing the processing when data is transferred to an I/O. This processing is performed by sequencer control as follows:

- ① The CPU starts the DMA controller and the I/O device.
- ② The DMA controller issues a request to acquire the system bus.
- ③ When the bus has been acquired, data transfer from the memory → data buffer memory is performed.
- ④ The DMA controller checks to see if the number of data bytes in the data buffer memory exceeds the number of valid bytes or not. If it is "yes", proceed to ⑥; if "no", return to ③.
- ⑤ The I/O device waits until the data request DREQ to the DMA controller is "on".
- ⑥ The DMA controller selects the I/O device data register port and performs a transfer of data from the data buffer memory to the I/O device.
- ⑦ Processes ⑤ and ⑥ continue until the data buffer memory is empty.
- ⑧ Check to see if the transfer counter is "0" or not; if it is "no", return to process ②.

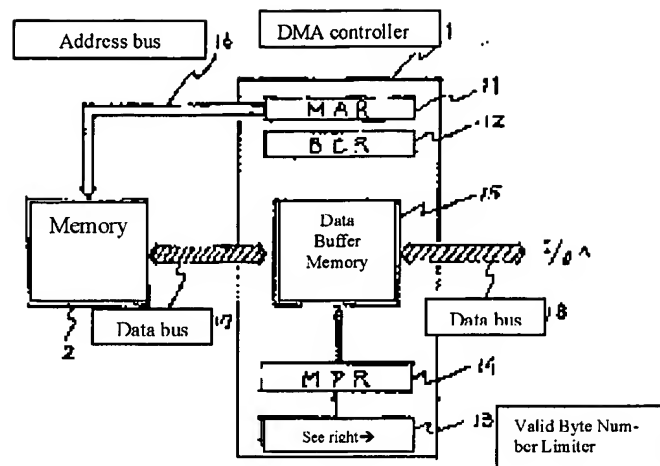
Effect of the Invention

As is clear from the foregoing description, the DMA controller of this invention has the function of taking transfer data into the internal data buffer memory during DMA transfers, and a burst DMA transfer is performed as a single data size data buffer memory. The difference between the present invention and the prior art lies in the fact that the valid byte size of the data buffer memory is variable, and the burst transfer time is variable.

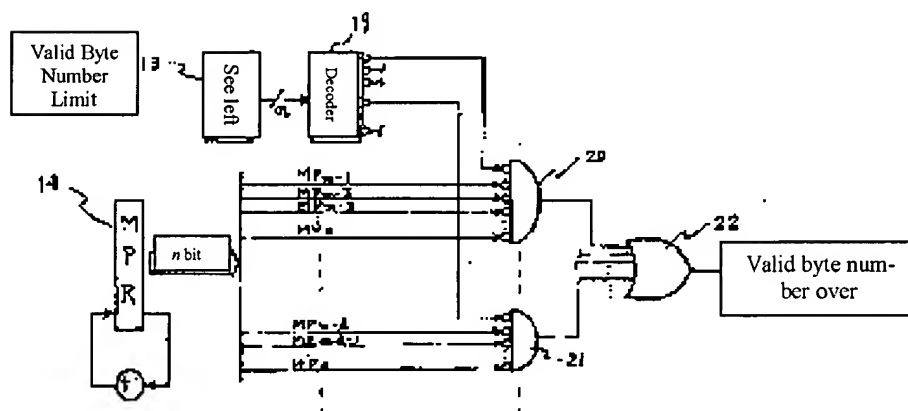
Brief Description of the Drawings

Figure 1 shows the DMA controller of this invention; Figure 2 is a block diagram of a configuration of a hardware mechanism for detecting the overage of valid bytes; Figure 3 is a block diagram of a configuration of a sequencer used in this invention; Figure 4 is a flow chart showing the processing when data is transferred from I/O to memory; Figure 5 is a flow chart showing the processing when data is transferred to an I/O; Figure 6 shows a prior art DMA control method.

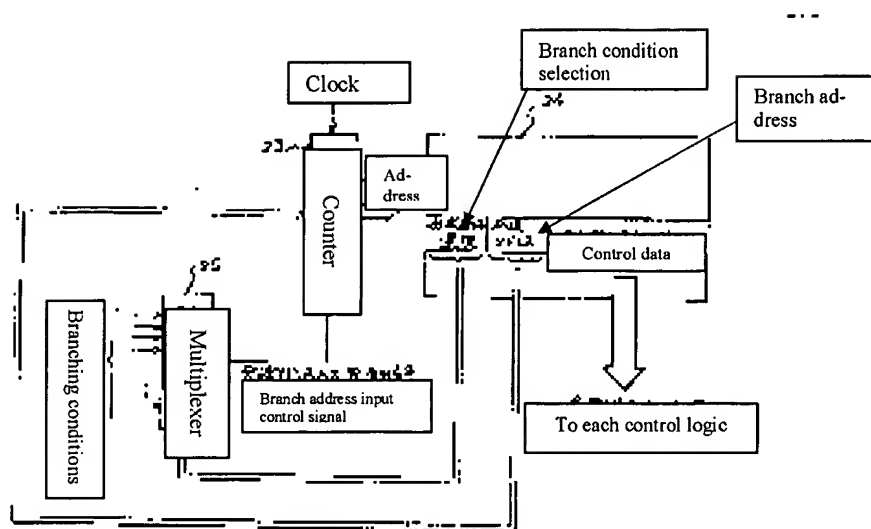
- 1 DMA controller
- 2 Memory
- 3 I/O device
- 4 CPU
- 11 Memory address register
- 12 Byte counter register
- 13 Valid byte number limiting register
- 14 Memory pointer register
- 15 Data buffer memory
- 17, 18 Data bus
- 19 Decoder
- 20, 21 AND circuit
- 22 OR circuit



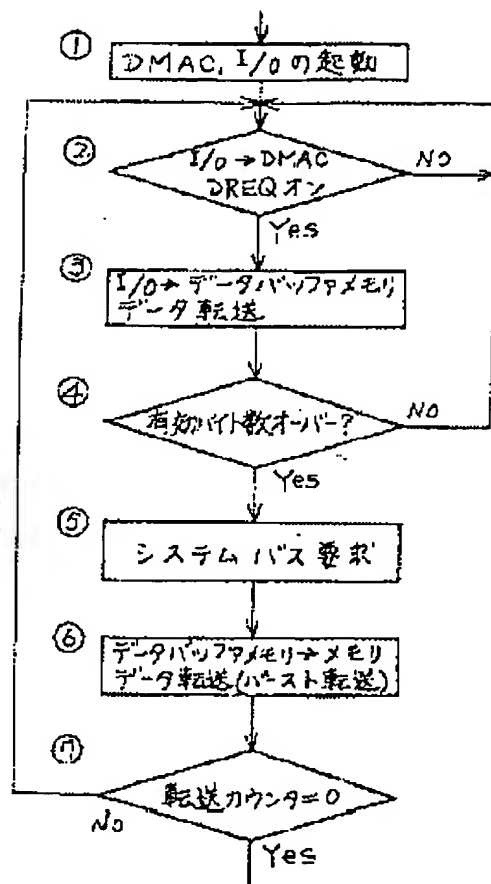
DMA Controller of this Invention
Fig. 1



Valid Byte Number Overage Detection
Fig. 2



Sequencer Configuration
Fig. 3



Start DMAC I/O

I/O → DMAC DREQ on

I/O → data buffer memory
data transfer

Has the number of valid bytes
been exceeded?

System bus request

Data buffer memory → Data
transfer (burst transfer)

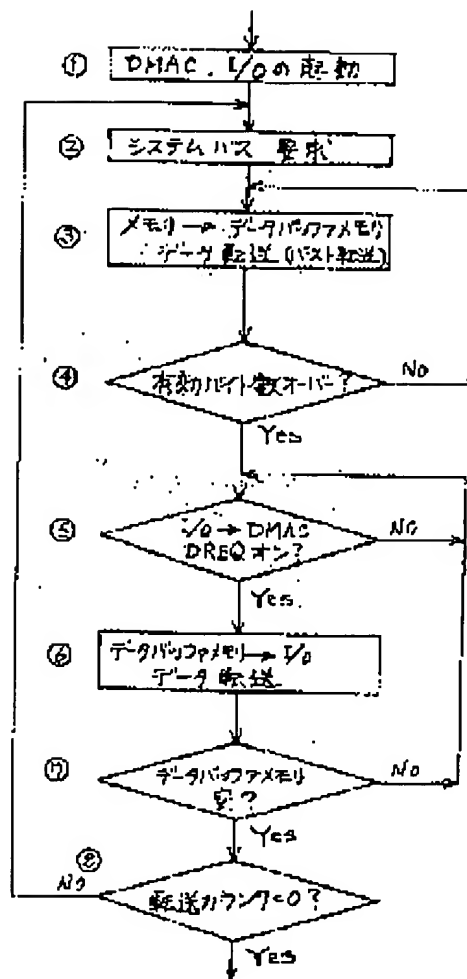
Transfer counter = 0

I/Oからメモリへのデータ転送

第4図

Memory data transfer from I/O

Fig. 4



Start DMAC I/O

System bus request

Memory → Data buffer memory data transfer (burst transfer)

Has the number of valid bytes been exceeded?

I/O → DMAC DREQ on?

Data buffer memory → I/O data transfer

Data buffer memory empty?

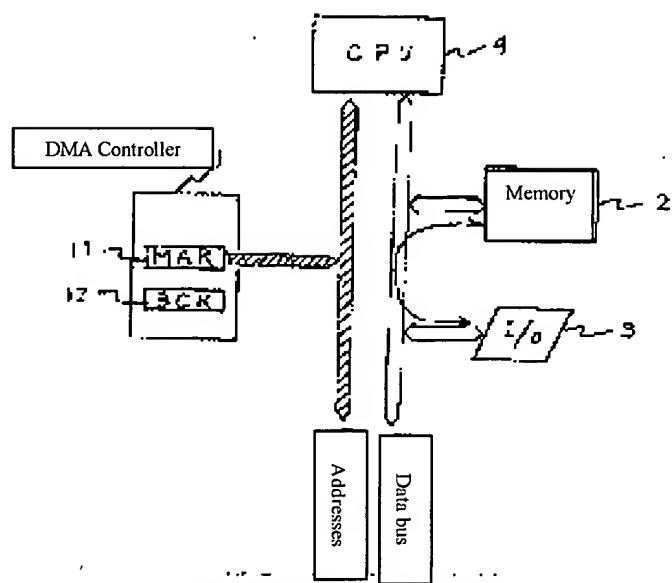
Transfer counter = 0?

メモリから I/O へのデータ転送

第 5 図

Data Transfer from Memory to I/O

Fig. 5



Prior Art DMA Control Method
Fig. 6

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☒ **BLACK BORDERS**

☒ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☒ **FADED TEXT OR DRAWING**

☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.